НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:
Завідувач кафедри
_____ Сергій СТІРЕНКО
«___»_____20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерні системи та мережі»
спеціальності 123 «Комп'ютерна інженерія»
на тему: «Система оцінки кібер ризиків в страхуванні»

Виконав:
студент IV курсу, групи ІО-63
Тіку Владислав Вячеславович _____

Керівник:
професор кафедри ОТ
Луцький Георгій Михайлович _____

Консультант з нормоконтролю:
Професор кафедри ОТ, д.т.н.
Сімоненко Валерій Павлович
_____

Рецензент доцент кафедри СПіСКС
к.т.н., доц.
Орлова Марія Миколаївна _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій СТІРЕНКО
«___»_____20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Тіку Владиславу Вячеславовичу

1. Тема проєкту «Система оцінки кібер ризиків в страхуванні», керівник проєкту Луцький Георгій Михайлович, професор кафедри ОТ, затверджені наказом по університету від «07» травня 2020 р. № 1081-с

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту див. технічне завдання

4. Зміст пояснювальної записки дослідження предметної області, огляд існуючих рішень, визначення вимог і завдань для програмного продукту, вибір платформи та технології, обґрунтування оптимальності використання обраних інструментів для розробки, реалізація проєкту.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко В.П.		

7. Дата видачі завдання _____

* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проєкту.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Відмітки про виконання
1	*Затвердження теми роботи*	01.09.2019	виконано
2	*Вивчення та аналіз завдання*	02.09.2019-02.02.2020	виконано
3	*Розробка архітектури та загальної структури систем*	03.02.2020-03.03.2020	виконано
4	*Розробка структур окремих Підсистем*	04.03.2020-15.03.2020	виконано
5	*Програмна реалізація системи*	16.03.2020-12.04.2020	виконано
6	*Оформлення пояснювальної записки*	13.04.2020-17.05.2020	виконано
7	*Захист програмного продукту*		
8	*Передзахист*	26.05.2020	
9	*Захист*		

Студент Владислав ТІКУ

Керівник Луцький Георгій Михайлович

Анотація

У бакалаврській дипломній роботі реалізовано метод збору та аналізу даних компаній.

Метод дозволяє пробудувати інтерактивну часову лінію та виставити можливу ціну страхової пропозиціх заснованих на даних з відкритих джерел. Програмний продукт був реалізований на мові Python за допомогою бібліотеки Beautifulsoup у візуальному середовищі PyCharm на хмарній платфомі AWS.

Аннотация

В бакалаврской дипломной работе реализован метод сбора и анализа данных компаний.

Метод позволяет пробудуваты интерактивную временную линию и выставить возможную цену страховой предложений основанные на данных из открытых источников. Программный продукт был реализован на языке Python с помощью библиотеки Beautifulsoup в визуальной среде PyCharm на облачной платфомы AWS.

Annotation

The method of data collection and analysis of companies is implemented in the bachelor's thesis.

The method allows you to build an interactive timeline and set the possible price of insurance offers based on data from open sources. The software was implemented in Python using the Beautifulsoup library in the PyCharm visual environment on the AWS cloud platform.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	А4		Завдання на дипломний проєкт	2	
2	А4	ІАЛЦ.467100.002 ТЗ	Система оцінки кібер ризиків у страхуванні Технічне завдання	4	
3	А4	ІАЛЦ.467100.003 ПЗ	Система оцінки кібер ризиків у Страхуванні. Пояснювальна записка	57	
4	А4	ІАЛЦ.467100.004 А1	Система оцінки кібер ризиків у Страхуванні. Схема структурна – структура програми	1	
5	А4	ІАЛЦ.467100.005 А2	Система оцінки кібер ризиків у Страхуванні. Схема функціональна – схема прецедентів	1	
6	А4	ІАЛЦ.467100.006 А3	Система оцінки кібер ризиків у Страхуванні. Схема принципова – схема алгоритму додавання нового пристрою	1	

Технічне завдання
до дипломного проєкту

на тему: «Система оцінки кібер ризиків в страхуванні»

Київ – 2020

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку системи оцінки кібер ризиків в страхуванні.

Область застосування: open source рішення для підвищення якості оцінки кібер ризиків для страхових компаній.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання розробки системи оцінки кібер ризиків в страхуванні, затвердженою кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний Інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проєкту є розробка системи оцінки кібер ризиків в страхуванні.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з комп’ютерних технологій, публікації в періодичних виданнях, публікації в Інтернеті за даним питанням.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- Розробка хмарного рішення для маштабування рішень пов’язаних з аналізом ринку ризиків в страхування;
- Розробка інтерфейсу для керування запитами;
- Розробка ботів, що сканують WWW для знаходження інформації, що можуть бути корисними для страхових компаній.

5.2. Вимоги до програмного програмного забезпечення

- Операційна система Linux
- Хмарна частина: AWS, EC2, RDS
- Спосіб передачі даних: API, JSON

6. ЕТАПИ РОЗРОБКИ

	Дата
Затвердження теми роботи	01.09.2020
Вивчення та аналіз завдання	02.09.2019-02.02.2020
Розробка архітектури та загальної структури систем	03.02.2020-03.03.2020
Розробка структур окремих підсистем	04.03.2020-15.03.2020
Програмна реалізація системи	16.03.2020-12.04.2020
Оформлення пояснювальної записки	13.04.2020-17.05.2020
Захист програмного продукту	
Передзахист	26.05.2020
Захист	

Пояснювальна записка
до дипломного проєкту

на тему: «Система оцінки кібер ризиків у страхуванні»

Київ – 2020

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС	Операційна система
Фреймворк	Інфраструктура програмних рішень, що полегшує розробку складних систем.
Python	Мова програмування Python
DOM	Об'єктна модель документа (англ. Document Object Model, DOM) — специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами
HTML	Мова розмітки гіпертекстових документів (англ. HyperText Markup Language) — стандартна мова розмітки веб-сторінок в Інтернеті.
API	Прикладний програмний інтерфейс(англ. Application Programming Interface, API)
JSON	Текстовий формат обміну даними між комп'ютерами, який дозволяє описувати об'єкти та інші структури даних(англ. JavaScript Object Notation)

ВСТУП

Сьогодні кібер-захист став однією з основних ідей ведення бізнесу. Тому багато компаній перш ніж стартувати новий продукт, думають про його захист. Окрім впровадження сучасних технологій захисту, бізнес охоче страхує свої продукти в у відповідних провайдерів провайдерів.

Актуальність теми

З кожним роком атаки на сервери компаній зростають. Хакери створюють все нові методи атак. Тому важко прогнозувати втрати та нові методи захисту . Основною причиної занепокоєння підприємців є або вразливість нульового дня. Тому компанії звертаються до страхових компаній, щоб вирівняти втрати, що обов'язковоо стануться. В свою чергу страхові аналітики проводять аудит для кліента, перевіряючи надійність архітектури, проводячи тренінги для співробітників. В залежності від оцінки аудиту, пропонується ціна страховки.

Часто сторона страхування намагається приховати можливі проблеми з безпекою: не якісне програмне забезпечення та обладнання, не достатнє фінансування віддіду захисту, не кваліфіковані спеціалісти, приховати всю можливу інформацію про взломи та втрати в інтернеті. Для чтого щоб отримати як умога дешеву страхову пропозицію.

В свою чергу страхові компанії намагаються проаналізувати всі аспекти в оціні кліента та виставити таку ціну страховку на яку кліент заслуговує. Тому що, вони ризикують виплатити повну страхову компенсацію.

Мета і задачі дослідження

Метою роботи є розробка методу аналізу потенційного кліента/компанії за допомогою інформації у відкритому доступі.

Для досягнення поставленої мети були поставлені наступні основні задачі:

- Провести аналіз існуючих систем аналізу відкритих даних;

- Створити програмну реалізацію розробленої системи;
- Провести тестування розробленої системи.

Практичне значення

Запропонована система оцінки кібер ризиків полягає у пошуку всієї можливої інформації в WWW, що може допомогти страховим компаніям чітко оцінити ціну страховки. Використання відкритих баз данних та API допоможе досягти високої якості аналізу даних та прогнозування.

РОЗДІЛ 1.
ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1. Загальні відомості

Система аналізу або CRA (Cybersecurity Risk Analysis) є частиною оцінки кібер ризиків (Cybersecurity Risk Assessment). Часто такі поняття збивають з пантелику.

Аналіз - процес уявного або справжнього розчленування складного об'єкта на частини для кращого розуміння [1]

Аналітика - основа інтелектуальної, логіко-мисленевої діяльності, спрямованої на рішення практичних завдань. У її основі лежить не стільки принцип констатації фактів, скільки принцип «випередження подій», що дозволяє організації або індивідові прогнозувати майбутній стан об'єкту аналізу. [2]

Оцінка - експертна оцінка проекту з метою визначення його прийнятності відповідно до прийнятих критеріїв. [3]

Ризик - це ймовірність можливої небажаної втрати чого-небудь при поганому збігу обставин. [4]

Всі ці поняття використовуються для страхування не тільки від кібер втрат/випадків і т.п.

За даними компанії McAfee, вона займається розробкою та реалізацією антивірусного програмного забезпечення, кібер-злочинці щорічно завдають світовій економіці збитків понад $600 млрд. Страховий концерн Lloyd's називає трохи іншу цифру – $400 млрд на рік.

1.1 Список різних методологій ризик-менеджменту

Фреймворк «NIST Risk Management Framework» на базі американських урядових документів NIST (National Institute of Standards and Technology, Національного інституту стандартів і технологій США) включає в себе набір

взаємозв'язаних т.зв. «Спеціальних публікацій» (англ. Special Publication (SP), будемо для простоти сприйняття називати їх стандартами):

- Стандарт NIST SP 800-39 «Managing Information Security Risk» («Управління ризиками інформаційної безпеки») пропонує трирівневий підхід до управління ризиками: організація, бізнес-процеси, інформаційні системи. Даний стандарт описує методологію процесу управління ризиками: визначення, оцінка, реагування та моніторинг ризиків.
- Стандарт NIST SP 800-37 «Risk Management Framework for Information Systems and Organizations» («Фреймворк управління ризиками для інформаційних систем і організацій») пропонує для забезпечення безпеки і конфіденційності використовувати підхід управління життєвим циклом систем.
- Стандарт NIST SP 800-30 «Guide for Conducting Risk Assessments» («Керівництво по проведенню оцінки ризиків») сфокусований на IT, ІБ і операційних ризиків. Він описує підхід до процесів підготовки і проведення оцінки ризиків, коммуніціровання результатів оцінки, а також подальшої підтримки процесу оцінки.
- Стандарт NIST SP 800-137 «Information Security Continuous Monitoring» («Безперервний моніторинг інформаційної безпеки») описує підхід до процесу моніторингу інформаційних систем і IT-середовищ з метою контролю застосованих заходів обробки ризиків ІБ і необхідність їх перегляду. 2. Стандарти Міжнародної організації зі стандартизації ISO (International Organization for Standardization):
- Стандарт ISO / IEC 27005: 2018 «Information technology - Security techniques - Information security risk management» («Інформаційна технологія. Методи і засоби забезпечення безпеки. Менеджмент ризику інформаційної безпеки») входить в серію стандартів ISO 27000 та є логічно взаємопов'язаним з іншими стандартами по ІБ з цієї серії. Даний

стандарт відрізняється фокусом на ІБ при розгляді процесів управління ризиками.

- Стандарт ISO / IEC 27102: 2019 «Information security management - Guidelines for cyber-insurance» («Управління інформаційною безпекою. Керівництво по кіберстрахуванію») пропонує підходи до оцінки необхідності придбання кіберстраховкі як заходи обробки ризиків, а також до оцінки і взаємодії зі страховиком.
- Серія стандартів ISO / IEC 31000: 2018 описує підхід до ризик-менеджменту без прив'язки до IT / ІБ. У цій серії варто відзначити стандарт ISO / IEC 31010: 2019 «Risk management - Risk assessment techniques» - на даний стандарт в його вітчизняному варіанті ДСТУ ISO / IEC 31010-2011 «Менеджмент ризику. Методи оцінки ризику »посилається 607-П ЦБ РФ« Про вимоги до порядку забезпечення безперебійності функціонування платіжної системи, показниками безперебійності функціонування платіжної системи та методикам аналізу ризиків у платіжній системі, включаючи профілі ризиків ».
- Методологія FRAP (Facilitated Risk Analysis Process) є відносно спрощеним способом оцінки ризиків, з фокусом тільки на найкритичніших активах. Якісний аналіз проводиться за допомогою експертної оцінки.
- Методологія OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) сфокусована на самостійній роботі членів бізнес-підрозділів. Вона використовується для масштабної оцінки всіх інформаційних систем і бізнес-процесів компанії.
- Стандарт AS / NZS 4360 є австралійським і новозеландським стандартом з фокусом не тільки на IT-системах, але і на бізнес-здоров'я компанії, тобто пропонує більш глобальний підхід до управління ризиками. Відзначимо, що даний стандарт зараз замінений на стандарт AS / NZS ISO 31000-2009.

- Методологія FMEA (Failure Modes and Effect Analysis) пропонує проведення оцінки системи з точки зору її слабких місць для пошуку ненадійних елементів.
- Методологія CRAMM (Central Computing and Telecommunications Agency Risk Analysis and Management Method) пропонує використання автоматизованих засобів для управління ризиками.
- Методологія FAIR (Factor Analysis of Information Risk) – пропріетарний фреймворк для проведення кількісного аналізу ризиків, що пропонує модель побудови системи управління ризиками на основі економічно ефективного підходу, прийняття поінформованих рішень, порівняння заходів управління ризиками, фінансових показників і точних ризик-моделей.
- Концепція COSO ERM (Enterprise Risk Management) описує шляхи інтеграції ризик-менеджменту зі стратегією і фінансової ефективністю діяльності компанії і акцентує увагу на важливість їх взаємозв'язку. У документі описані такі компоненти управління ризиками, як стратегія і постановка цілей, економічна ефективність діяльності компанії, аналіз і перегляд ризиків, корпоративне управління і культура, а також інформація, комунікація та звітність.

1.3. Що таке кіберстрахування?

Кіберстрахування - це спеціальний страховий продукт, призначений для захисту бізнесу та осіб, які надають послуги для таких підприємств, від ризиків, що базуються на Інтернеті, і загалом від ризиків, пов'язаних з інфраструктурою інформаційних технологій, інформаційною конфіденційністю, відповідальністю за управління інформацією та пов'язаними з нею заходами.

Такі ризики, як правило, виключаються з традиційних комерційних полісів загальної відповідальності або, принаймні, конкретно не визначені у традиційних страхових продуктах. Покриття, що надається полісами кіберстрахування, може включати покриття сторонніми сторонами втрат,

таких як знищення даних, вимагання, крадіжки, злому та відмова в нападі на послуги; покриття відповідальності, що відшкодовує підприємствам збитки для інших, спричинені, наприклад, помилками та упущеннями, невдачею щодо захисту даних або наклепом; та інші вигоди, включаючи регулярний аудит безпеки, зв'язки з громадськістю після інцидентів та слідчі витрати та грошові винагороди.

1.4. Переваги

Оскілька риноку кіберстрахування у багатьох країнах порівняно невеликий порівняно з іншими страховими продуктами, його загальний вплив на виникаючі кіберзагрози важко оцінити. Оскільки вплив кіберзагроз для людей та підприємств також є відносно широким порівняно із сферою захисту страхових продуктів, страхові компанії продовжують розвивати свої послуги. По мірі того, як страховики виплачують кіберзбитки, а кіберзагрози розвиваються та змінюються, страхові продукти все частіше купуються разом із існуючими службами IT-безпеки. Дійсно, критерії андеррайтингу для страховиків, що пропонують продукти кіберстрахування, також на початку розвитку, і андеррайтери активно співпрацюють з компаніями з безпеки IT для розробки своєї продукції. Окрім прямого покращення безпеки, кіберстрахування є надзвичайно вигідним у разі масштабного порушення безпеки.

Страхування забезпечує плавний механізм фінансування для відновлення великих втрат, допомагаючи бізнесу повернутися до нормального стану та зменшуючи потребу в державній допомозі. Нарешті, страхування дозволяє справедливо розподіляти ризики кібербезпеки, при цьому вартість премій пропорційна розміру очікуваних збитків від таких ризиків. Це дозволяє уникнути потенційно небезпечних концентрацій ризику, одночасно запобігаючи вільному їзду.

1.5. Недоліки

Інформаційні технології є властивою практично всім сучасним бізнесам, вимога до окремого продукту існує лише завдяки навмисному виконанню

заходів, які виключали крадіжки та збитки, пов'язані із сучасними технологіями, з існуючих ліній продуктів. Брюс Шнайер постулював, що існуючі страхові практики, як правило, дотримуються або моделі "Повені чи Пожежі", проте кіберподії не моделюються жодним із цих типів подій, це призвело до ситуації, коли сфера дії Кіберу Далі обмежується страхування, щоб зменшити ризик для андеррайтерів.

Це ускладнює недостатність даних, що стосуються фактичної шкоди, пов'язаної з типом події, відсутність стандартів, пов’язаних із класифікацією подій, та відсутність доказів, пов'язаних з ефективністю "найкращих практик у галузі".

Страхування покладається на обґрунтовані актуарні дані на тлі великого статичного ризику. Зважаючи на те, що їх наразі не існує, малоймовірно, що або покупці цих товарів досягнуть бажаних результатів. Цей погляд на ринок відображається в поточному стані ринку, коли стандартні виключення призводять до ситуації, коли "страховик може стверджувати, що вони застосовуються майже до будь-яких порушень даних".

1.6. Сучасні технології аналізу ризиків в інформаційних системах

Метою аналізу ризиків, пов'язаних з експлуатацією інформаційних систем (ІС), є оцінка загроз (т. Е. Умов і факторів, які можуть стати причиною порушення цілісності системи, її конфіденційності, а також полегшити несанкціонований доступ до неї) і вразливостей (слабких місць в захисту, які уможливлюють реалізацію загрози), а також визначення комплексу контрзаходів, що забезпечує достатній рівень захищеності ІС. При оцінюванні ризиків враховуються багато факторів: цінність ресурсів, значимість загроз, вразливостей, ефективність наявних і запланованих засобів захисту і багато іншого. Сучасні технології аналізу ризиків в Росії використовуються порівняно рідко. Основна причина такого становища полягає в тому, що в керівних документах (РД Держтехкомісії) не розглядаються аспект ризиків, їх допустимий рівень і відповідальність за прийняття певного рівня ризиків.

Інформаційна система, в залежності від свого класу, повинна мати підсистемою безпеки з конкретними формальними властивостями.

Аналіз ризиків, як правило, виконується формально, з використанням довільних методик. У розвинених країнах це не так. Наприклад, в американському глосарії з безпеки можна знайти термін Designated Approving Authority - особа, уповноважена прийняти рішення про допустимість певного рівня ризиків. Питанням аналізу ризиків приділяється серйозна увага: десятиліттями збирається статистика, удосконалюються методики. Проте нинішнє становище починає змінюватися.

Серед вітчизняних фахівців служб інформаційної безпеки (ІБ) зріє розуміння необхідності проведення такої роботи. В першу чергу це відноситься до банків і великим комерційним структурам, т. Е. До тих, які серйозно піклуються про безпеку своїх інформаційних ресурсів.

1.6.1. Основні підходи до аналізу ризиків

В даний час використовуються два підходи до аналізу ризиків - базовий і повний варіант. Вибір залежить від оцінки власниками цінності своїх інформаційних ресурсів і можливих наслідків порушення режиму інформаційної безпеки.

У найпростішому випадку власники інформаційних ресурсів можуть не оцінювати ці параметри. Мається на увазі, що цінність ресурсів з точки зору організації не є надмірно високою. У цьому випадку аналіз ризиків проводиться за спрощеною схемою: розглядається стандартний набір найбільш поширених загроз без оцінки їх ймовірності і забезпечується мінімальний або базовий рівень ІБ. Повний варіант аналізу ризиків застосовується в разі підвищених вимог до ІБ. На відміну від базового варіанту в тому чи іншому вигляді оцінюються ресурси, характеристики ризиків і вразливостей. Як правило, проводиться аналіз співвідношення вартість / ефективність декількох варіантів захисту.

Таким чином, при проведенні повного аналізу ризиків необхідно:

- визначити цінність ресурсів;

- додати до стандартного набору список загроз, актуальних для досліджуваної інформаційної системи;

- оцінити ймовірність загроз;

- визначити вразливість ресурсів;

- запропонувати рішення, що забезпечує необхідний рівень ІБ.

1.6.2. Визначення цінності ресурсів

Ресурси зазвичай поділяються на кілька класів - наприклад, фізичні, програмні ресурси, дані. Для кожного класу необхідна своя методика визначення цінності елементів, що допомагає вибрати відповідний набір критеріїв. Ці критерії служать для опису потенційного збитку, пов'язаного з порушенням конфіденційності і цілісності ІС, рівня її доступності.

Фізичні ресурси оцінюються з точки зору вартості їх заміни або відновлення працездатності. Ці вартісні розміри потім перетворюються в рангову (якісну) шкалу, яка використовується також для інформаційних ресурсів. Програмні ресурси оцінюються тим же способом, що і фізичні, на основі визначення витрат на їх придбання або відновлення. Якщо для інформаційного ресурсу існують особливі вимоги до конфіденційності або цілісності (наприклад, якщо вихідний текст має високу комерційну цінність), то оцінка цього ресурсу проводиться за тією ж схемою, т. Е.

У вартісному вираженні. Крім критеріїв, що враховують фінансові втрати, комерційні організації можуть застосовувати критерії, що відображають:

- збиток репутації організації;

- неприємності, пов'язані з порушенням чинного законодавства;

- збиток для здоров'я персоналу;

- збиток, пов'язаний з розголошенням персональних даних окремих осіб;

- фінансові втрати від розголошення інформації;

- фінансові втрати, пов'язані з відновленням ресурсів;

- втрати, пов'язані з неможливістю виконання зобов'язань;

- збиток від дезорганізації діяльності.

Можуть використовуватися й інші критерії залежно від профілю організації. Наприклад, в урядових установах вдаються до критеріїв, що відображає специфіку національної безпеки і міжнародних відносин

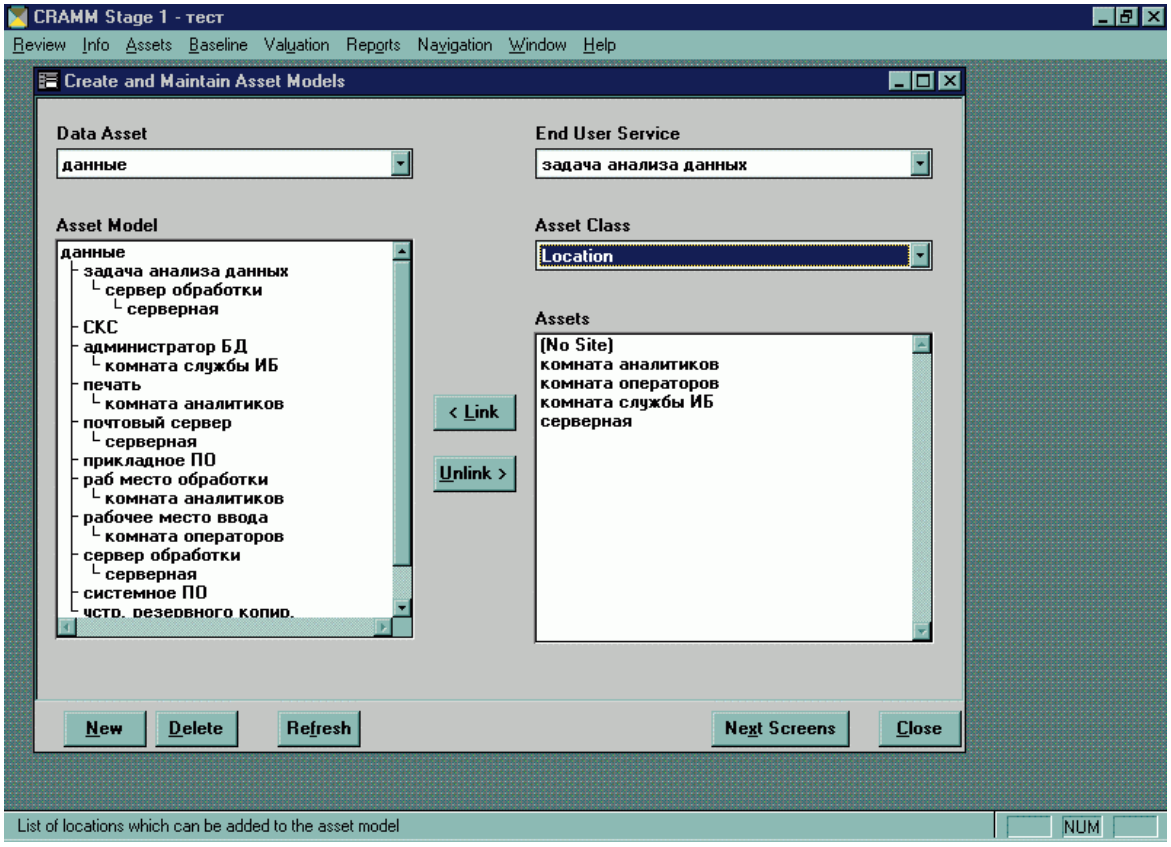


Рис. 1.1 Процедура оцінки ризиків CRAMM

1.6.3. Типічні сценарії

Крадіжка даних

- Реквізити банківських рахунків клієнтів, їх карт та інші персональних дані були вкрадені.
- Необхідно провести лекції для клієнтів, моніторинг їх рахунків, можливі вимоги від постраждалих клієнтів

Недоступність даних

- Вірус Петя шифрує вміст даних жорстких дисків на всіх десктопах і лептопах компаній

- Робота компанії зупиняється на 2 тиждні (або більше) поки всі машини заміняються або відновлюються.

Порушення роботи системи

- Хакер проникає в систему, що керує роботою обладнаня та місцях.
- Відбуваються збої в операційній діяльності з причини не можливості контроля віддалених об’єктів.

Атаки на Автоматизовані системи керування технологічним процесом

- Вірус накшталт Stuxnet заражає систему підприємницького типу.
- Хакери отримують контроль над ключевими об’єктами та обладнянням контроля тиску, що призводить до порушення режиму роботи і серйозному розриву нафтопродуктів.

1.7 Провайдери кібер ризиків та потенційних користувачів системи

- **AXA**

Один з найбільших провайдерів страхування. AXA пропонує широкий спект аудиту, експертизи та консалтингу в цій сфері.

- Проактивне управління ризиками: В рамках полісів кіберстрахування AXA пропонуєм клієнтам проактивні інструменти, послуги та ресурси для виявлення, пом’якшення та реагування на кіберзагрози.
- Гнучке покриття: пропонує гнучкі кібер-продукти, які охоплюють конфіденційність, мережу, засоби масової інформації, помилки та упущення тощо.
- Претензії, орієнтовані на клієнтів: Спеціалізована команда з претензій готова допомогти. Вони співпрацюють з клієнтами для вирішення порушень кібербезпеки, швидкого реагування, відновлення та продовження вашого бізнесу вперед. Про претензії можна повідомляти 24 години на день, сім днів на тиждень.

AIG

Кіберстрахування AIG можна записати через автономну політику CyberEdge® або затвердити на вибраних полісах фінансових ліній, майна та випадкових випадків. Перегляньте наш посібник із кіберпрофілю, щоб ознайомитись із найкращими підходами для вашої компанії. [3]

- Детальна оцінка загрози та аналітика.
- Страхувальники отримують детальну оцінку, аналіз та звітні показники, які допоможуть їм краще зрозуміти свою кіберзрілість та покриття *. Основні звіти також доступні для заявників, навіть якщо вони не пов'язують покриття.
- Інструменти та послуги з попередження втрат
- Кіберстрахувальники озброєні широким спектром інструментів та послуг - вартістю до 25 000 доларів США - включені у відповідні поліси, щоб допомогти забезпечити додатковий захист від викупу, запобігти працівникам не стати жертвою фішинг-атак тощо.
- Глобальна експертиза претензій
- Після виклику на «гарячу лінію» команда претензій CyberEdge координує свою діяльність з клієнтом для виконання плану дій реагування, залучає будь-яких необхідних постачальників, включаючи адвокатів з порушення та криміналістичних фірм для виявлення негайних загроз (наприклад, хакера всередині мережі) та запускає процеси відновлення та відновлення.

1.7 Ланцюги Маркова

Ланцюг Маркова в математиці це випадковий процес, що задовольняє властивість Маркова і який приймає скінченну чи зліченну кількість значень (станів). Існують ланцюги Маркова як з дискретним так і з неперервним часом. В даній статті розглядається дискретний випадок.

Марковські моделі складаються з вичерпних уявлень можливих ланцюгів подій, тобто переходів, всередині систем, які у разі аналізу надійності та доступності відповідають послідовностям відмов і ремонту.

- Стан подій та переходів

Модель Маркова вивчає ймовірність перебування в заданому стані в даний момент часу, кількість часу, яку система повинна провести в заданому стані, а також очікувану кількість переходів між станами. Наприклад, це корисно при розгляді несправностей та ремонту.

- Переваги

Марковські моделі дозволяють детально представити процеси відмов та відновлення, особливо якщо це стосується залежностей. Аналіз Маркова добре підходить для обробки рідкісних подій, на відміну від аналізів, заснованих на симуляції, і тому дозволяє аналізувати такі події протягом розумного часу. Марковські моделі також можуть застосовуватися в будь-якій ситуації, коли відомі окремі стани та переходи між ними. Іноді ці стани явні протилежності, такі як "Робота" проти "Невдало" або "Хороша" проти "Погана", але в більшості випадків між ними є багато станів, які також можна пояснити за допомогою Марківських моделей.

- Застосування

Аналіз Маркова - це техніка, яка використовується для отримання чисельних заходів, пов'язаних з вірогідністю заданого стану, надійністю та наявністю системи або частини системи. Аналіз Маркова виконується, коли залежності між відмовою кількох компонентів, а також залежностями між

відмовами компонентів і коефіцієнтами відмов не можуть бути легко представлені за допомогою комбінації дерев несправностей та інших методик.

Algorithm 1

Calculate stationary probabilities $\pi_1 = p21/(p12+p21)$, $\pi_2 = 1-\pi_1$
Generate a uniformly distributed random number $u \in U(0,1)$
if $u \leq \pi_1$ **then**
    $s_1 = 1$
else
    $s_1 = 2$
end if
for $i = 2$ to $i = N$ **do**
    Generate a uniformly distributed random number $u \in U(0,1)$
    if $s_{i-1} = 1$ AND $u \leq p_{12}$ **then**
        $s_i = 2$
    else if $s_{i-1} = 2$ AND $u \leq p_{21}$ **then**
        $s_i = 1$
    else
        $s_i = s_{i-1}$
    end if
end for

Рис. 1.2. Алгоритм моделювання для 2-стану ланцюга Маркова.

1.8 Процес управління кібер ризиками

Страхування полягає в управлінні певним сприйнятим ризиком. Купуючи медичне страхування, ви маєте ризик захворіти. Купуючи автомобільне страхування, ви керуєте ризиком нещасного випадку. Кіберстрахування не відрізняється: купуючи кіберстрахування, ви керуєте ризиком інциденту, пов'язаного з кібербезпекою. Але тут закопане питання: скільки страховки вам потрібно? Ви не єдиний, хто цікавиться цим питанням: страхові компанії постійно задають собі це питання, адже це в кінцевому підсумку має все значення. Слід прийняти за очевидне, що необхідна сума страхування пропорційна величині ризику, якому ви схильні. Страхові компанії намагаються оцінити ризик за допомогою аналізу ризику, який вони

постійно вдосконалюють. Вони знають, які ключові фактори, що визначають ризик.

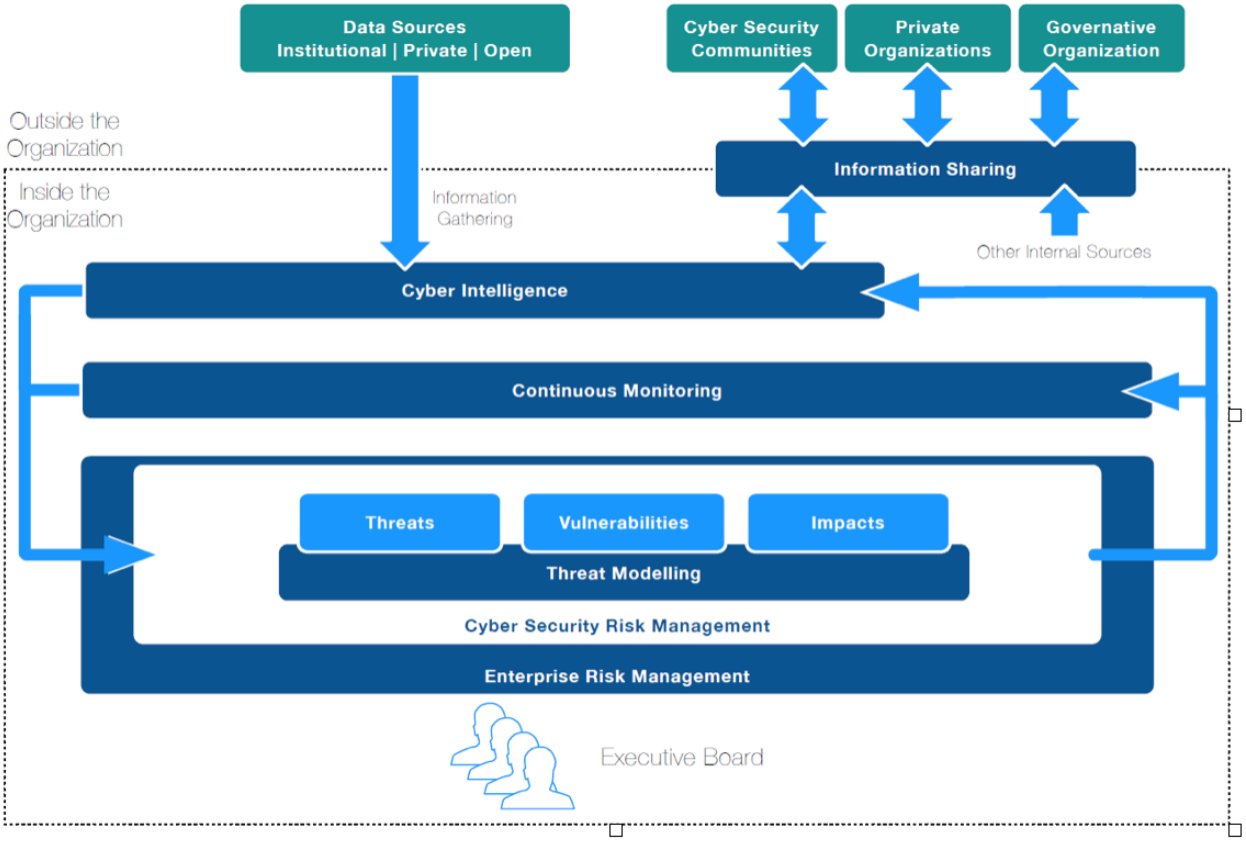


Рис. 1.3 Процес управління ризиками в кібербезпеці

ВИСНОВОК ДО РОЗДІЛУ 1

1. У даному розділі було проаналізовано методи оцінки ризиків.

2. Було проаналізовано різновиди систем аналізу ринку, їх переваги та недоліки.

3. Проаналізували математичні моделі та алгоритми на яких будується аналіз кібер ризиків не вдалось, так як ця інформація є конфіденційною. Розриття таких даних вплине на репутацію компанії.

РОЗДІЛ 2.
ПРОЄКТУВАННЯ ДОДАТКУ

2.1 Опис предметної області

В даній бакалаврській роботі буде спроектована система збору необроблених даних та аналітика. Основна задача цієї системи є пошук відповідних даних, про потенційною кліента.

Побудова time-line для компаній, що були взломані або понесли втрати від інших факторів через не надійність їх систем.

2.1.1 Хмарні обчислення

Хмарні технології (або хмарні обчислення, cloud computing) - технології розподіленої обробки цифрових даних, за допомогою яких комп'ютерні ресурси надаються інтернет-користувачеві як онлайн-сервіс. Програми запускаються і видають результати роботи в вікні web-браузера на локальному ПК. Основною перевагою використання хмарних обчислень є маштабування.

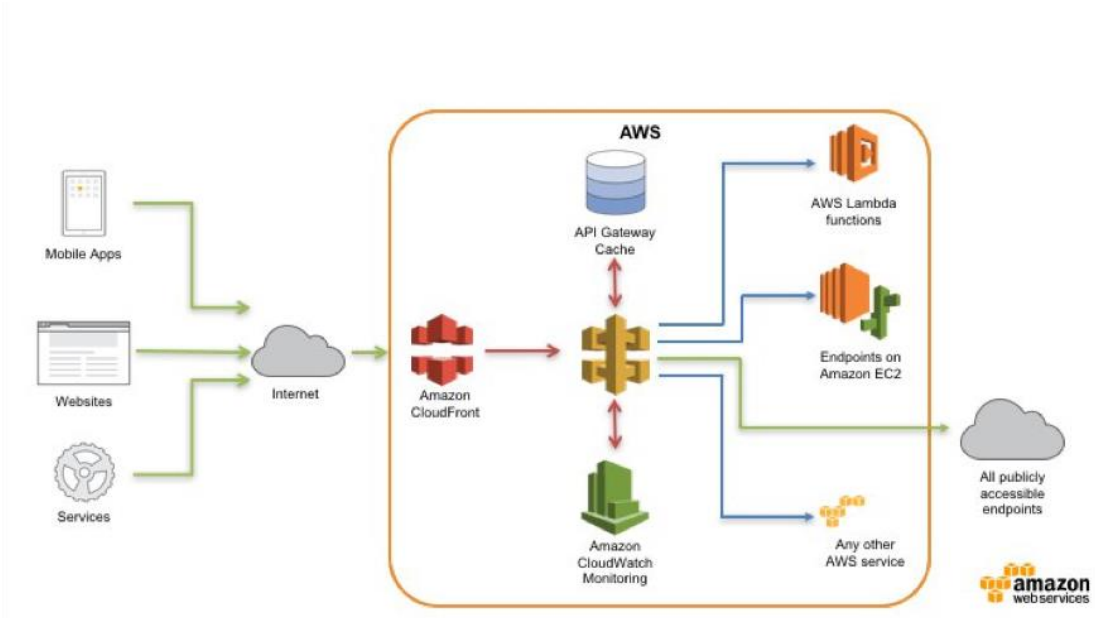


Рис. 2.1. Типова схема взаємодії веб сайтів та інших сервісів.

Хмарні обчислення - це доступ на замовлення ресурсів комп'ютерної системи, особливо для зберігання даних (хмарного сховища) та обчислювальної потужності, без безпосереднього активного управління

користувачем. Термін зазвичай використовується для опису центрів обробки даних, доступних багатьом користувачам через Інтернет. Великі хмари, що переважають сьогодні, часто мають функції, розподілені по декількох місцях від центральних серверів. Якщо з'єднання з користувачем є відносно близьким, він може бути призначений крайовим сервером. [2]

Хмарні обчислення набагато абстрактніші як рішення для віртуального хостингу. Замість того, щоб бути доступними через фізичне обладнання, усі сервери, програмне забезпечення та мережі розміщуються в хмарі, поза приміщеннями. Це віртуальне середовище в реальному часі, яке розміщується між декількома різними серверами одночасно. Тож замість того, щоб вкладати гроші в придбання власних фізичних серверів, ви можете орендувати простір для зберігання даних у постачальників хмарних обчислень на більш економічній основі за плату за використання.

2.1.2 Інтернет бот

Інтернет-бот, веб-робот, робот або просто бот - це програмне забезпечення, яке виконує автоматизовані завдання (сценарії) через Інтернет. Зазвичай боти виконують завдання, прості та повторювані, набагато швидше, ніж людина могла. Найбільш широке використання ботів - для веб-сканування, в якому автоматизований скрипт отримує, аналізує та зберігає інформацію з веб-серверів. Більше половини всього веб-трафіку генерується ботами.

2.1.3 Python Google-Search

Python бібліотека для скрепінгу гугл запитів. Ця бібліотека використовується для пошуку інофрмації. Скажімо, ви працюєте над проектом, який потребує веб-вискоблювання, але ви не знаєте веб-сайтів, на яких потрібно заздалегідь виконати скребки, замість цього вам потрібно виконати пошук у Google, а потім перейти до результатів пошуку Google на кількох веб-сайтах. У такому випадку вам потрібен результат пошуку Google для ваших різних запитів. Один із способів досягти цього - використання

запиту та прекрасного супу, про який вже йшлося в Розділі з впровадження веб-скрапінгу на Python з BeautifulSoup. Замість того, щоб докласти стільки зусиль для тривіального завдання Google, було зроблено пакет. Це майже одне вкладене рішення для прямого пошуку посилань усіх результатів пошуку Google. Використовуючи пакет пакунків python google, ми можемо отримати результат пошуку Google із скрипту python. Ми можемо отримати посилання перших n результатів пошуку.нтернеті.

2.1.4 Бібліотека BeautifulSoup

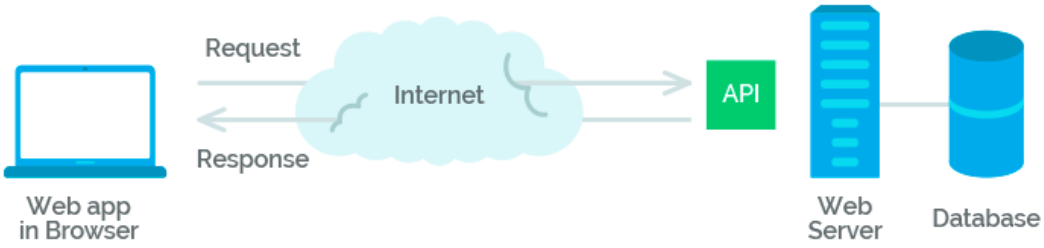
Beautiful Soup - бібліотека Python для витягу даних з HTML та XML-файлів. Він працює з вашим улюбленим аналізатором, щоб забезпечити ідіоматичні способи навігації, пошуку та зміни дерева аналізу. Зазвичай це економить програмістам години або дні роботи.

2.1.5 Принцип роботи Парсера

Парсер - це програма або пошукова система (граббер або скрипт), що проводить аналіз інформації сторінок Інтернет-сайтів. Вона організовує збір даних (парсит) і структурує її. Парсер проводить синтаксичний аналіз текстової інформації з математичної моделі, за якою порівнюються лексеми з формальної граматикою. Аналогічно можна описати дію людини при читанні слів, тобто лексем. Він теж проводить синтаксичний аналіз, тобто порівняння прочитаних слів з тими, що є в його словниковому запасі або формальної граматикою.

2.1.6 Інтерфейс програмування додатків

API (інтерфейс програмування додатків) - це сукупність правил і механізмів, за допомогою яких одна програма або компонент взаємодіє з іншими. Здається, назва говорить сама за себе, але давайте заглибимось у деталі. API може повернути потрібні для вашої програми дані у зручному форматі (наприклад, JSON або XML). У цьому підручнику з RESTful API ми зупинимося лише на JSON.



Типова схема роботи API

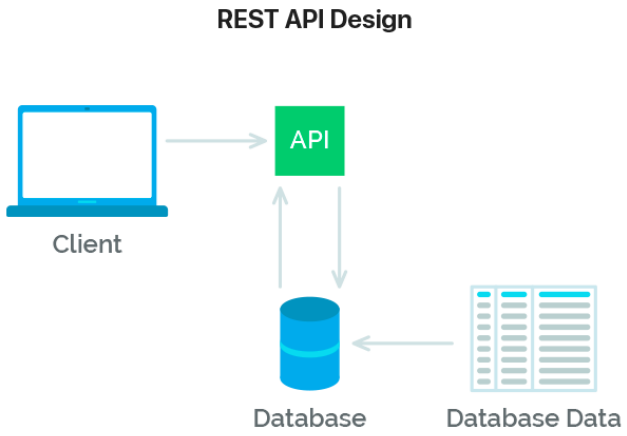


Рис. 2.2. Дизайн REST API

Незважаючи на те, що в процесі створення API існує безліч інструментів та технологій, популярними інструментами та продуктами для розробки API для розробників є:

- Apigee: це постачальник програм управління API Google, який допомагає розробникам та підприємцям перемагати в цифрову трансформацію шляхом відновлення до підходу API.
- APIMatic і API-трансформатор: Вони пропонують складні інструменти автоматичного покоління для складання SDK-кодів вищої якості та фрагментів коду з конкретних форматів API та перетворення їх в інші специфікаційні форми, такі як RAML, API Blueprint тощо.
- API Science: Цей інструмент використовується в першу чергу для оцінки продуктивності як внутрішніх, так і зовнішніх API.

- Архітектура без сервера API: Ці продукти допомагають розробникам мобільних додатків розробляти, створювати, публікувати та розміщувати API за допомогою хмарної серверної інфраструктури.
- API-платформа: це одна з PHP-систем з відкритим кодом, яка підходить для розробки веб-API.
- Auth0: це рішення управління ідентифікацією, яке використовується для аутентифікації та авторизації API.
- ClearBlade: це постачальник управління API для втілення технології IoT у ваш процес.
- GitHub: Цей сервіс хостингу репозиторію git з відкритим кодом дозволяє розробникам керувати файлами коду, витягувати запити, контролювати версії та коментувати, які поширюються по всій групі. Це також дозволяє їм зберігати свій код у приватних сховищах.
- Листоноша: Це, в основному, ланцюжок інструментів API, яка надає розробникам можливість запускати, тестувати, документувати та оцінювати ефективність їх API. Хоча у світі повно API та API, все ще існує великий розрив у використанні переваг API. У той час як деякі API роблять інтеграцію до програми легким вітром, інші перетворюють це на кошмар. Щоб допомогти вам у створенні попереднього, ось деякі фактори, пов’язані з API, які розробники повинні врахувати:

2.1.7 Веб-сервіси Amazon

Amazon Web Services (AWS) є дочірньою компанією Amazon, яка надає платформи та API для хмарних обчислень на замовлення фізичним особам, компаніям та урядам на основі дозованої оплати. У сукупності ці веб-сервіси хмарних обчислень забезпечують набір примітивної абстрактної технічної інфраструктури та розподілених будівельних блоків та інструментів. Один з таких сервісів - Amazon Elastic Compute Cloud (EC2), який дозволяє користувачам мати в своєму розпорядженні віртуальну кластерну мережу комп'ютерів, доступну весь час через Інтернет. Версія віртуальних комп’ютерів AWS емулює більшість атрибутів реального комп'ютера,

включаючи апаратні центральні процесорні блоки (процесори) та графічні одиниці обробки (GPU) для обробки; локальна / оперативна пам'ять; накопичувач на жорсткому диску / SSD; вибір операційних систем; мережа; попередньо завантажене прикладне програмне забезпечення, таке як веб-сервери, бази даних та управління відносинами з клієнтами (CRM). Технологія AWS впроваджена на серверних фермах у всьому світі та підтримується дочірньою компанією Amazon. Плата заснована на поєднанні використання (відомого як модель "оплата за переходом"), апаратних / ОС / програмних / мережевих функцій, обраних абонентом, необхідної доступності, надмірності, безпеки та можливостей обслуговування. Абоненти можуть платити за один віртуальний комп'ютер AWS, виділений фізичний комп'ютер або кластери будь-якого. В рамках угоди про підписку Amazon забезпечує безпеку для систем абонентів. AWS працює з багатьох глобальних географічних регіонів, включаючи 6 у Північній Америці. У 2020 році AWS містила понад 212 сервісів, включаючи обчислення, зберігання, мережу, базу даних, аналітику, сервіси прикладних програм, розгортання, управління, мобільний, інструменти для розробників та інструменти для Інтернету речей. До найпопулярніших можна віднести EC2 та Amazon Simple Storage Service (Amazon S3). Більшість сервісів не піддаються впливу кінцевих користувачів, а натомість пропонують функціональні можливості через API, які розробники можуть використовувати у своїх додатках. Пропозиції Amazon Web Services доступні через HTTP, використовуючи архітектурний стиль REST та протокол SOAP для старих API та виключно JSON для новіших. Amazon продає AWS абонентам як спосіб отримати великі масштаби обчислювальної техніки швидше та дешевше, ніж побудова фактичної ферми фізичних серверів. Усі послуги тарифікуються на основі використання, але кожна служба вимірює використання різними способами. Станом на 2017 рік AWS володіє домінуючим 34% усієї хмари (IaaS, PaaS), тоді як наступні три конкуренти Microsoft, Google та IBM мають 11%, 8%, 6% відповідно відповідно до Synergy Group [1].

2.3. Визначення вимог і завдань

Основними функціями системи є:

1. Пошук інформації в WWW про компанію/установу:
   a. За певний період;
   b. На конкретних ресурсах;
   c. Глибина пошуку;
   d. Аналіз інформації
2. Керування запитами за допомогою REST API;
3. Авторизація;
4. Налаштування пошуку;

Основними вимогами до системи є:

1. Фільт пошуку;
2. API взаємодія;
3. TimeLine ;
4. Локалізація системи англійською мовою;
5. Система повинна бути доступним з будь-якої точки світу.

2.4. Опис функціоналу системи

Система орієнтована для страхових компаній, що займаються страхуванням від кібер атак.

Функціонал для страхових компаній:

1. Отримання унікального токена;
2. Робити запит по конкретній компанії;
3. Налаштування фільтрів пошуку;
4. Зберігати результат в PDF форматі;
5. Перегляд time line;

ВИСНОВОК ДО РОЗДІЛУ 2

У розділі 2 було проведено опис предметної області проєкту, складені основні функції та вимоги до функціоналу додатку. Також були розроблені можливі сценарії прецедентів під час виконання основних функцій системи та визначені можливі виключні ситуації, які можуть негативно впливати на функціональність роботи системи. Це дозволить користувачу максимально комфортно використовувати систему.

Було проаналізовано хмарні рішення а також методи парсингу. Найкращим варіантом для побудови системи є хмарний сервіси. Так як вони є лідером в цій сфері та пропонують дуже зручну та маштабовану архітектуру.

Спосіб обміну інформацією був визначений за API. Тому що це найефективніший спосіб передачі даних, а також є маштабованим. Використовуючи HTTPS протокол можна результувати, що метод є ще й найбезпечнішим.

РОЗДІЛ 3.
РОЗРОБКА ДОДАТКУ

3.1. Вибір технологій та їх обґрунтування

3.1.1. Вибір платформи для додатку

У розділах 1 та 2 були визначені вимоги щодо проектування системи оцінки кібер ризиків та були поставлені такі завдання:

1) Система має працювати на RESTfull архітектурі.
2) Можливість налаштування фільтрів пошуку.
3) Система повинен працювати швидко в залежності від фільтрів.
4) Цілодобовий моніторинг кліентів на рахунок взломів або down time серверів.

Для того, щоб усі вимоги були дотримані, необхідно обрати платформу для розробки. Вона повинна бути зручна та швидка, щоб зробити систему якомога якіснішою. Серед запропонованих систем було виявлено такі найбільш популярні мобільні та комп'ютерні платформи:

1) Операційні системи Windows, Linux, macOS;
2) Хмарні провайдери: AWS, Azure, Google Cloud.

Для розробки системи була вибрана операційна система Ubuntu Server 18.04 LTS. Це популярна операційна система для серверної частини з підтримкою мінімум 8 років. Широкий набір інструментів, швидкість та простота в використанні, робить її найкращим варіантом.

3.1.2. Вибір мови програмування

Для розробки API частини, пошуку відновідних ресурсів та аналізу оберемо мову програмування Python.

Мова програмування Python останнім часом все частіше використовується для аналізу даних, як в науці, так і комерційній сфері. Цьому сприяє простота мови, а також велика різноманітність відкритих бібліотек

3.1.3. Вибір допоміжних бібліотек

Sklearn - бібліотека, алгоритмів машинного навчання, вона знадобиться нам надалі для класифікації досліджуваних даних,

Matplotlib - бібліотека для побудови графіків.

Pandas - бібліотека для обробки і аналізу даних. Будемо використовувати для первинної обробки даних.

Numpy - математична бібліотека з підтримкою багатовимірних масивів,

Google-translate - бібліотека для перекладу тексту, через google API (для використання потрібно отримати API ключ в Google).

Pycountry - бібліотека, яку будемо використовувати для перетворення коду країни в повну назву країни.

Основні рішення з реалізації системи та його компонентів

Розробку додатку можна поділити на такі пункти:

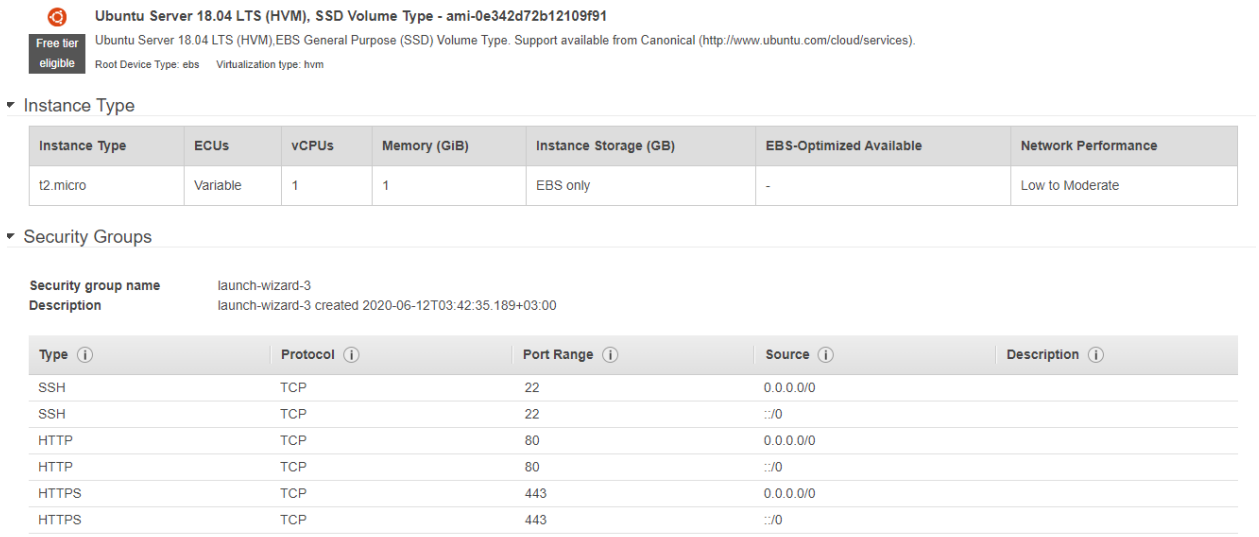
- Налаштування хмарної частини
- Реалізація кліентської частини
- Реалізація серверної частини
- Реалізація API
- Створення бази даних
- Реалізація бота
- Тестування

3.2 Основні рішення з реалізації додатку та його компонентів

3.2.1 Налаштування хмарної частини

На цьому етапі проводиться налаштування сервера.

- Вибір операційної системи (Ubuntu Server 18.04 LTS)
- Об’єм оперативної пам’яті (1 GiB memory)
- Об’єм вбудованої пам’яті (25 GiB SSD)
- Вибір процесора (Intel Xeon Family)
- Налаштування фаерволу (SSH, HTTP,HTTPs)



Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0e342d72b12109f91

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name launch-wizard-3
Description launch-wizard-3 created 2020-06-12T03:42:35.189+03:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
HTTPS	TCP	443	0.0.0.0/0	
HTTPS	TCP	443	::/0	

Рис. 3.1 Налаштування сервера

3.2.2 Ініціалізація та налаштування бази даних

Для зберігання інформації зібраної за допомогою бота використовується PostgreSQL.

- PostgreSQL 11.6-R1
- 1 vCPU
- 1 GiB RAM

Перший крок процедури інсталяції - налаштувати вихідне дерево для вашої системи та вибрати параметри, які б вам хотілося. Це робиться за допомогою запуску сценарію налаштування.

./configure

Цей скрипт виконає ряд тестів для визначення значень для різних змінних, що залежать від системи, та виявлення будь-яких химерностей вашої операційної системи, і нарешті створить кілька файлів у дереві збірки для запису того, що він знайшов .

Ви також можете запустити налаштування в каталозі поза деревом-джерелом, якщо ви хочете зберігати каталог збірки окремо. Ця процедура також називається збіркою VPATH.

mkdir build_dir cd build_dir / path / to / source / tree / configure [options go here]

gmake

Конфігурація за замовчуванням створємо утиліти, а також усі клієнтські програми та інтерфейси, для яких потрібен лише компілятор C. Усі файли за замовчуванням будуть встановлені під

/ usr / local / pgsql.

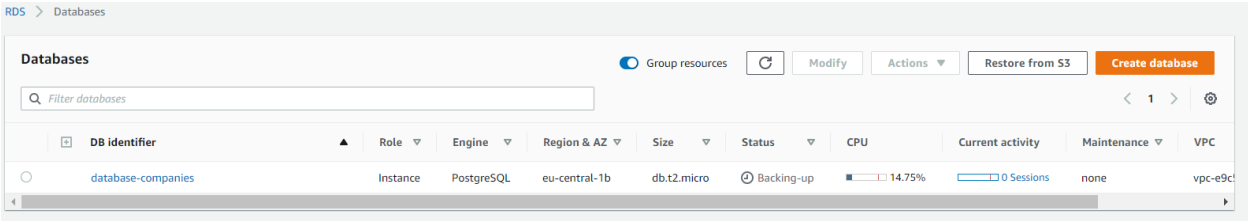


Рис. 3.2 Налаштування бази даних

3.2.3 Створення сховища для зберігання великих за розміром даних

Це сховище буде використовуватися для зберігання даних про компанії не за запитом страхової компанії.

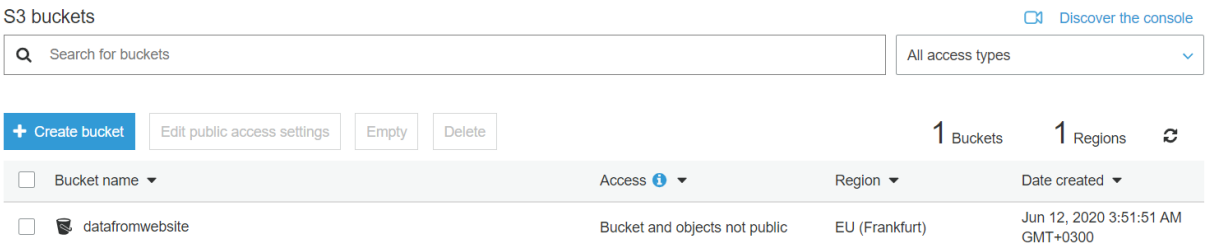


Рис. 3.3. Створення сховища

3.2.4 Створення клієнтської частини

Для реалізації клієстської частини було використано фреймворк Django

Django - це вільна веб-рамка, заснована на Python, що відповідає архітектурній схемі модельного виду-шаблону (MVC). Його підтримує Django Software Foundation (DSF), американська незалежна організація, створена як неприбуткова. Основна мета Django - полегшити створення складних веб-сайтів, керованих базами даних. Рамка наголошує на повторному використанні та «підключеності» компонентів, меншій кількості коду, низькому з'єднанні, швидкому розвитку та принципі не повторювати себе . Python використовується на всьому протязі, навіть для файлів налаштувань та моделей даних. Django також надає додатковий адміністративний інтерфейс для створення, читання, оновлення та видалення, який генерується динамічно за допомогою самоаналізу та конфігурується через адміністраторські моделі.

Основний функціонал - це введення назви компанії/потенційного кліента та отримання EXCEL документу/репорту.

Рис. 3.4. Клієнтська частина

3.2.5 Реалізація серверної частини

Для побудови сервеної архітектури ми використовуєм стек інструментів LAMP.

LAMP (Linux, Apache, MySQL, PHP / Perl / Python) - дуже поширений приклад стека веб-служб, названий як абревіатура імен його первинних чотирьох компонентів з відкритим кодом: операційна система Linux, сервер Apache HTTP , система управління реляційними базами даних MySQL (RDBMS) та мова програмування PHP. Компоненти LAMP багато в чому взаємозамінні і не обмежуються оригінальним вибором. Як пакет рішень, LAMP підходить для створення динамічних веб-сайтів та веб-додатків. З часу свого створення модель LAMP була адаптована до інших компонентів, хоча, як правило, складається з безкоштовного програмного забезпечення з відкритим кодом. Наприклад, еквівалентна установка в родині операційних систем Microsoft Windows відома як WAMP, а еквівалентна установка на macOS відома як MAMP.

З цього стеку нам знадобляться такі основні компоненти:

Apache - HTTP-сервер Apache, розмовно називаний Apache, - це вільне та відкрите джерело крос-платформного веб-сервера, випущене за умовами ліцензії Apache 2.0. Apache розробляється та підтримується відкритою спільнотою розробників під егідою Software Apache Software Foundation.

Server version: Apache/2.4.29 (Ubuntu)
Server built: 2020-03-13T12:26:16

Python - інтерпретована мова програмування високого рівня, загального призначення. Створена Гідо ван Россумом та вперше випущена в 1991 році, філософія дизайну Python підкреслює читабельність коду завдяки помітному використанню значного пробілу. Його мовні конструкції та об'єктно-орієнтований підхід мають на меті допомогти програмістам написати чіткий логічний код для малих та масштабних проектів

Для зберігання репортів та даних про компанії а також список компаній використовується PostgreSQL, окремим сервісом RDS на платформі AWS.

PostgreSQL, також відомий як Postgres, це безкоштовна та відкрита система управління реляційними базами даних (RDBMS), що підкреслює розширюваність та відповідність SQL. Спочатку вона мала назву POSTGRES, посилаючись на своє походження як спадкоємця бази даних Інгрес, розробленої в Каліфорнійському університеті, Берклі. У 1996 році проект було перейменовано на PostgreSQL, щоб відобразити його підтримку SQL. Після огляду в 2007 році команда розробників вирішила зберегти ім'я PostgreSQL та псевдонім Postgres.

```
sudo psql -h database-companies.######.eu-central-1.rds.amazonaws.com -p 5432 -U ubuntu
```

3.2.6 Метод пошуку інформації в інтернеті

Простий додаток Search Console за допомогою Python, який отримує список сайтів, підтверджених у вашому обліковому записі консолі пошукової консолі, та перелічує всі подані файли мапи сайту. Для запуску прикладу швидкого запуску вам знадобиться:

- Доступ до Інтернету та веб-браузера, щоб авторизувати зразок програми.
- Обліковий запис Google з принаймні одним веб-сайтом, підтвердженим у пошуковій консолі Google.

- Середовище для запуску програм на вибраній мові.

3.2.7 Метод парсингу вебсайтів

Складність парсингу потенційних сайтів, що містять інформацію про атаки на компанію є індивідуальними. Тобто парсинг кожного сайту потрібно проводити написанням нових скриптів, що ускладнює збір даних. В майбутній розробці буде використовуватися AI, що буде знаходити опитимальні новини та сервіси, на яких знаходиться інформація про клієнта. Для реалізації використаємо наступні веб сервіси. За допомогою них побудуємо таблицю даних та запишем їх в базу даних, а потім на клієнстську частину в EXCEL.

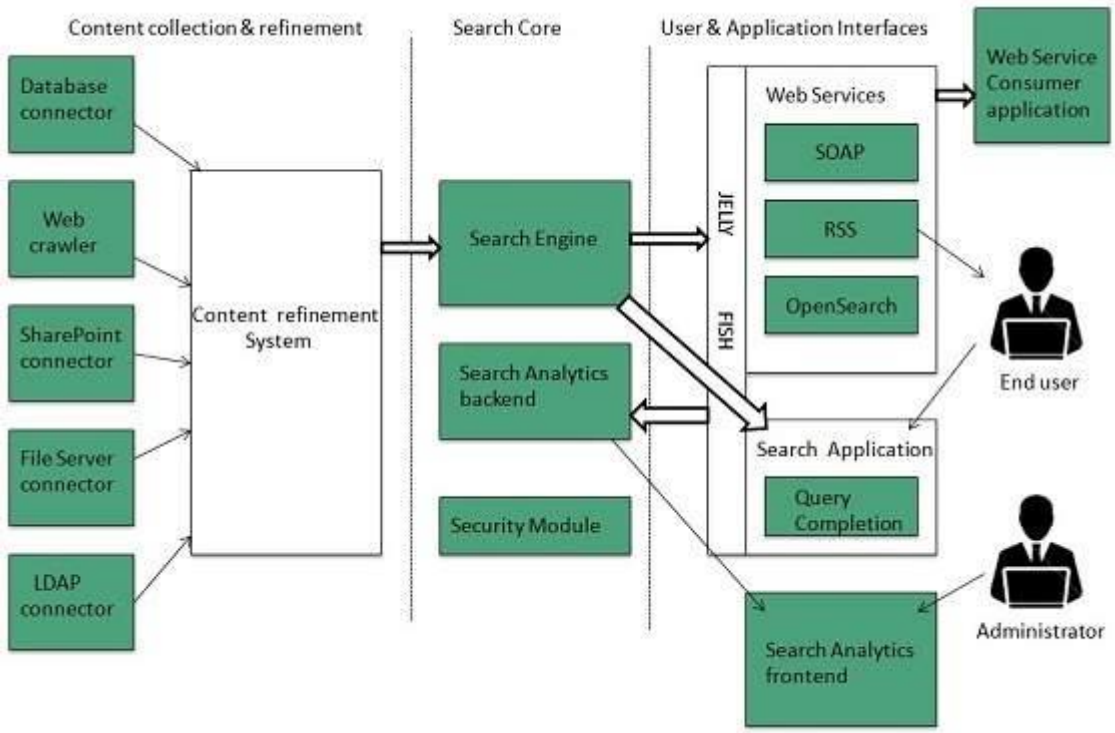


Рис. 3.5. Типова схема роботи пошукової архітектури Google

Таблиця 3.1 – Джерела парсингу

Державні установи	https://usr.minjust.gov.ua/ua/freesearch	Юридична інформація
Соціальні медія	https://www.linkedin.com/notifications/ https://dou.ua/ https://www.facebook.com/ rabota.ua/	Соціальна інформація. Аналізу коментарів та постів компанії. А також моніторинг ваканцій.
Пошукові двигуни	google.com https://www.yahoo.com/	Пошук всієї можливої інформації
Джерела новин	https://www.unian.ua/ https://tsn.ua/ https://www.ukr.net/ua/ https://itukraine.org.ua/news/	Пошук за новинам. Моніторинг сайти для перехоплення новини, що пов’язані з кліентом
Веб сайт кліента	http://uber.com/	Аналіз структури сайту. Нових постів, каканцій, мітапів, хакатонів, партнерів.
Сайти цінних паперів	https://www.bloomberg.com/europe https://gotoshop.ua/uk/kiev/ https://gerchik.ru/stati/	Моніторинг цін акцій, капіталізації
RSS	https://eng.uber.com/	Пошук новин за RSS

3.3 Json структура

Реалізувавши парсер та проаналізувавши дані, можемо будувати JSON структуру. Даний формат як найкраще підходить для описання. Основні пункти, що потрібно описати в ньому:

- Загальна інформаці про компанію Uber
- Фінансова частина
- Пошук ваканцій компанією
- Timeline, що описує основні зміни в компанії а також кібер атаки та дії що могли бути причиною фінансових втрат.
- Партнери
- Оцінка захищенності
- Можливі ризики під час страхування

3.3.1 Типовий приклад

В даній структурі описується основні положення по компанії

"Company name": "Uber Technologies Inc",

"General Information": {

"Stock price": "$32.24",

"Value": "+1.14 (+3.67%)",

"CEO": "Dara Khosrowshahi (Aug 30, 2017–)",

"Founded": "March 2009, San Francisco, California, United States",

"Number of employees": "22,263 worldwide, including 11,488 outside the United States (2018)",

"Revenue": "14.15 billion USD",

"Subsidiaries": [

"Uber Eats",

```
    "Careem",
    "Jump",
    "Otto",
    "deCarta",
    "MORE"
  ],
  "Founders": [
    "Garrett Camp",
    "Travis Kalanick"
  ]
 },
 "Search for specialists": {
  "Linkedin": {
    "Engineering Manager": 3,
    "Driver": 4,
    "Food Delivery Driver (131)": 131,
    "Senior Software Engineer": 5,
    "Software Engineer": 5
  }
 },
 "TimeLine": [
  {
    "year": "2009",
```

```
  "month and date": "March",
  "event type": "Company",
  "details": "Uber founded as UberCab."
 },
 {
  "year": "2010",
  "month and date": "May",
  "event type": "Company",
  "details": "Uber goes live for the first time in San Francisco"
 },
 {
  "year": "2010",
  "month and date": "December",
  "event type": "Team",
  "details": "Ryan Graves steps down as CEO in favor of Travis Kalanick.[4]"
 },
 {
  "year": ""
 },
 {
  "year": "2011",
  "month and date": "February 14",
```

"event type": "Funding",

"details": "Uber announces it has raised $11 million in Series A round led by Benchmark Capital."

},

{

"year": ""

},

{

"year": "2011",

"month and date": "May",

"event type": "National expansion",

"details": "Uber goes live in New York City"

}

ВИСНОВОК ДО РОЗДІЛУ 3

У даному розділі був проведений аналіз технологій оцінки кібер ризиків, а також огляд додаткових бібліотек, які можуть допомогти вирішити поставленні задачі. Враховуючи усі переваги та недоліки кожної технології, було обрано мову програмування,плаформу та фреймворк, що допоможе реалізувати такий додаток – Django, AWS, Python.

Реалізацію системи та його компонентів було розбито на такі основні етапи: створення хмарної архітектури; реалізація пошукового бота; налаштування баз даних; оцінка ризиків. Кожен етап реалізації виконувався згідно вимог, зазначених у технічному завданні. У результаті система була побудована як web-application, але такод може використовуватися як API

Були наведені відповідні рисунки готової сторінки Cyber Security Insurance Manager та API.

Висновки

Розробка даного дипломного проєкту була присвячена дослідженню можливих варіантів рішень, спрямованих на реалізацію системи оцінки кібер ризиків.

У ході виконання проєкту були розглянуті вже існуючі на даний момент рішення, які реалізують системи оцінки. На основі цих рішень було складено порівняльну характеристику з урахуванням всіх переваг та недоліків кожної системи.

Також було проаналізовано предметну область даної роботи та визначено основні вимоги і функції додатку, а також наведений список прецедентів та різних сценаріїв функцій, які можуть зустрітися у системі. На основі визначених прецедентів та сценаріїв були побудовані схематичні рисунки та таблиці. Проведено проєктування інтерфейсу та побудовано відповідні макети сторінки системи та API.

Зважаючи на вище задані вимоги був проведений аналіз використання існуючих технологій та платформ для реалізації системи. Веб-версія буде реалізована майже у всіх можливих браузерах та відображатися практично однаково. Також було обрано мову написання проєкту, а також проведений опис використаних додаткових бібліотек з обґрунтуванням доцільності їх використання.

Реалізація системи відбувалася у 5 основних етапів з урахуванням зазначеного у технічному завданні функціоналу та вимог до розробки, а саме реалізацію системи оцінки кібер ризиків.

Список використаної літератури

1. AWS [Електронний ресурс] – Режим доступу до ресурсу: https://aws.amazon.com/training/learning-paths/
2. Python Doc [Електронний ресурс] – Режим доступу до ресурсу: https://docs.python.org/2/library/parser.html
3. Google Chrome [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Google_Chrome
4. HTML [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/HTML
5. Cyber Insurance [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Cyber_insurance
6. Markov Models [Електронний ресурс] – Режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-3-319-43742-2_24
7. AWS architecture [Електронний ресурс] – 2020 – Режим доступу до ресурсу: https://www.tutorialspoint.com/amazon_web_services/amazon_web_services_basic_architecture.htm
8. RESTfull architecture [Електронний ресурс] – Режим доступу до ресурсу: https://restfulapi.net/rest-architectural-constraints/#:~:text=REST%20Architectural%20Constraints,the%20development%20of%20web%20services.
9. What is API [Електронний ресурс] – Режим доступу до ресурсу: https://www.mulesoft.com/resources/api/what-is-an-api
10. Load Balancer [Електронний ресурс] – Режим доступу до ресурсу: https://www.f5.com/services/resources/glossary/load-balancer#:~:text=A%20load%20balancer%20is%20a,users)%20and%20reliability%20of%20applications.

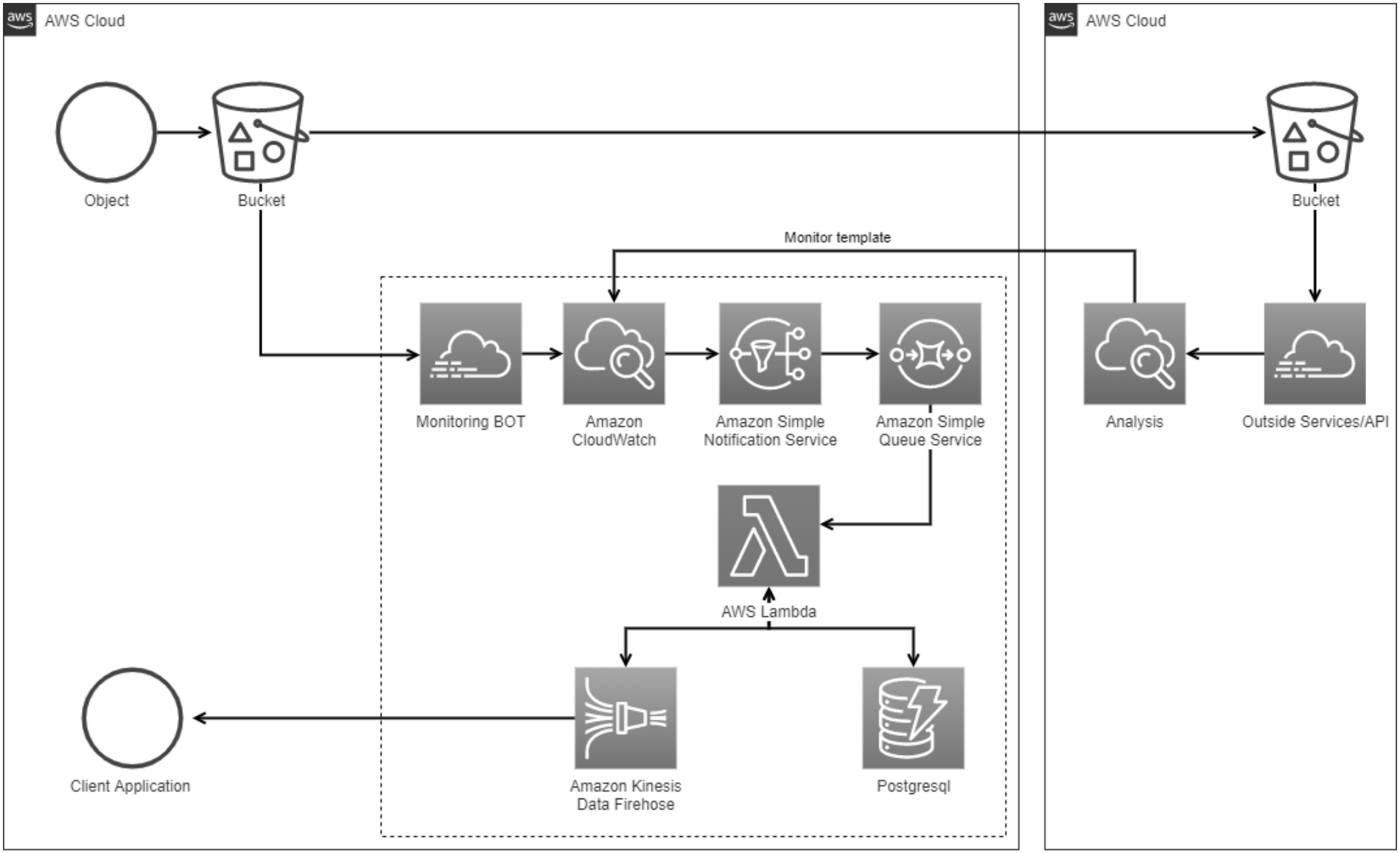
11. Cyber strategy for insurers [Електронний ресурс] – Режим доступу до ресурсу: https://www.ey.com/Publication/vwLUAssets/ey-cyber-strategy-for-insurers/$File/ey-cyber-strategy-for-insurers.pdf
12. Markov chains in insurance [Електронний ресурс] – Режим доступу до ресурсу: https://people.kth.se/~armerin/FinInsMathRwanda/Lecture19.pdf
13. JSON Web Token [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/JSON_Web_Token
14. Web application [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Web_application

ДОДАТОК 1
Система аналізу кібер ризиків

Схема структурна – структура програми
ІАЛЦ.467100.004 Д1

Аркушів 1

Київ – 2020 р.



					ІАЛЦ.467100.004 Д1
Зм.	Арк.	№ докум.	Підпис	Дата	
Розробив		Тіку В.В.			Система оцінки кібер ризиків у страхуванні. Схема структурна
Перевірив		Луцький Г.М.			
Реценз.					
Н. Контр.		Сімоненко В.П.			
Затв.		Стіренко С.Г.			

Літ.	Аркуш	Аркушів
	3	1

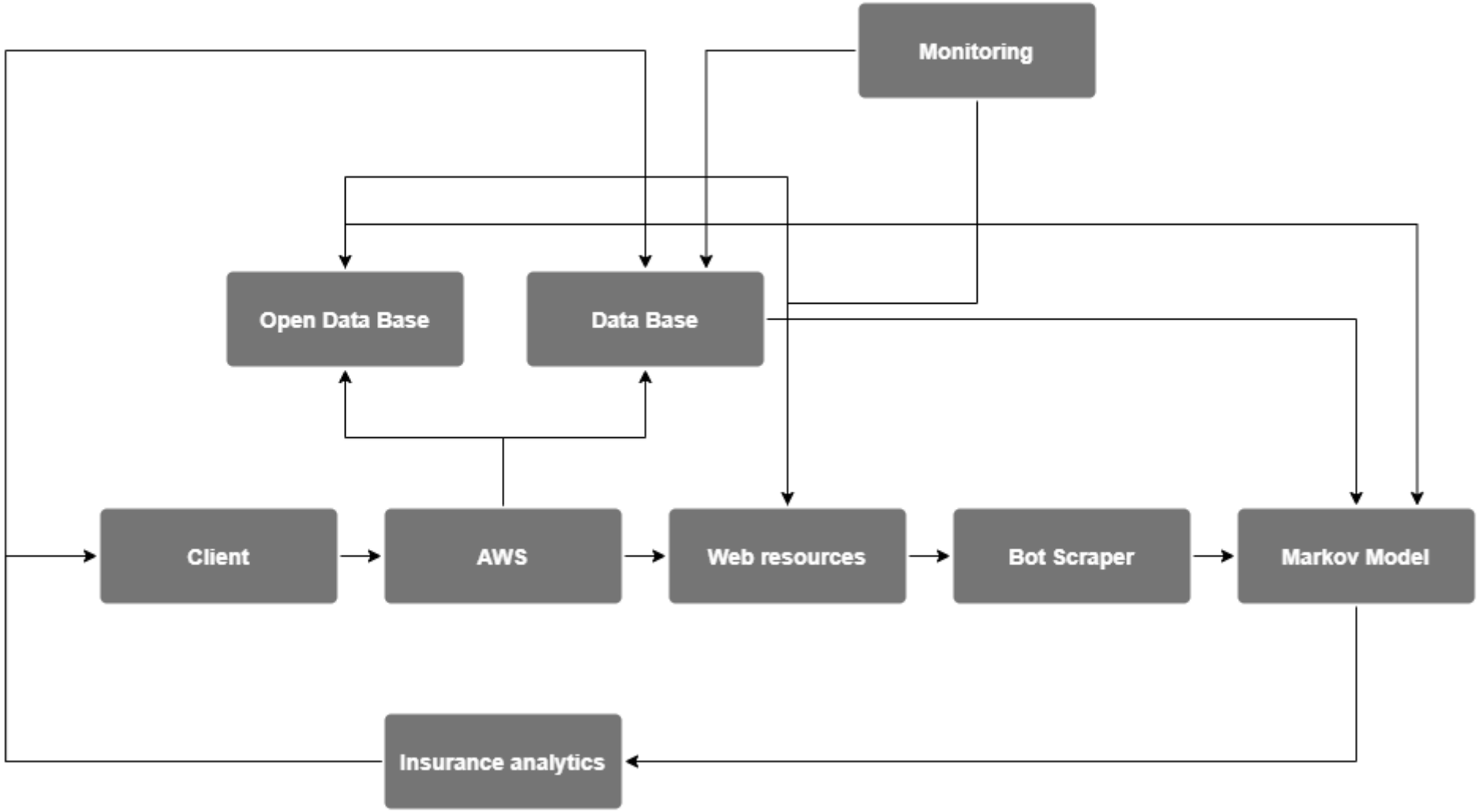
НТУУ «КПІ», ФІОТ, ІО-63

ДОДАТОК 2
Система аналізу кібер ризиків

Схема функціональна – схема прецедентів
ІАЛЦ.467100.005 Д2

Аркушів 1

Київ – 2020



					ІАЛЦ.467100.005 Д2
Зм.	Арк.	№ докум.	Підпис	Дата	
Розробив		Тіку В.В.			Система оцінки кібер ризиків у страхуванні. Схема функціональна
Перевірив		Луцький Г.М.			
Реценз.					
Н. Контр.		Сімоненко В.П.			
Затв.		Стіренко С.Г.			

Літ.	Аркуш	Аркушів
	3	1

НТУУ «КПІ», ФІОТ, ІО-63

ДОДАТОК З

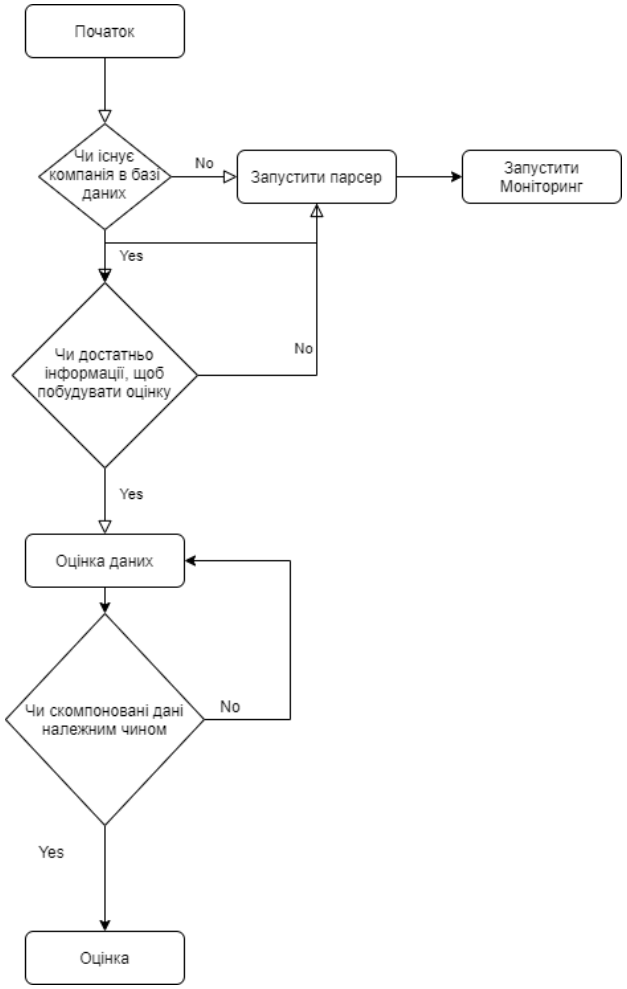
Система аналізу кібер ризиків

Схема принципова – схема алгоритму додавання нового пристрою

ІАЛЦ.467100.006 ДЗ

Аркушів 1

Київ – 2020



					ІАЛЦ.467100.006 Д3
Зм.	Арк.	№ докум.	Підпис	Дата	
Розробив		Тіку В.В.			Система оцінки кібер ризиків у страхуванні. Схема принципова
Перевірив		Луцький Г.М.			
Реценз.					
Н. Контр.		Сімоненко В.П.			
Затв.		Стіренко С.Г.			

Літ.	Аркуш	Аркушів
	5	1

НТУУ «КПІ», ФІОТ, ІО-63